日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　１月１７日

出願番号
Application Number: 特願２００３－０１０４５４

[ST. 10/C]: [ＪＰ２００３－０１０４５４]

出願人
Applicant(s): ソフトウェア生産技術研究所株式会社

２００３年１０月１０日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫

出証番号　出証特２００３－３０８３６４３

【書類名】　特許願
【整理番号】　2002-109
【提出日】　平成15年 1月17日
【あて先】　特許庁長官殿
【発明者】
　【住所又は居所】　東京都大田区南雪谷４丁目２２番１４号
　【氏名】　平山　貞宏
【発明者】
　【住所又は居所】　神奈川県鎌倉市十二所９６７－６４
　【氏名】　根来　文生
【特許出願人】
　【識別番号】　599086238
　【氏名又は名称】　ソフトウェア生産技術研究所株式会社
　【代表者】　根来　文生
【代理人】
　【識別番号】　100110559
　【弁理士】
　【氏名又は名称】　友野　英三
【手数料の表示】
　【予納台帳番号】　164782
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　要件単語の変更方法並びに要件単語の新規規定方法

【特許請求の範囲】

【請求項１】ソフトウェアの要件として規定される要件単語（＝データ項目）群に係る、

該単語の名前、

該単語に対応する値を得る計算式（インプットによって値が得られることを含む。）、

該単語に対応する値が成立するための条件、

該単語がインプットかアウトプットかの別、

該単語の存在する記録媒体

で規定される情報を、規定された要件単語の配列の順序を問わないで自動プログラミングができる手法に適用して自動的に作成されるプログラムの要件の変更にあたり、

（ａ）変更すべき要件単語自身の規定を変更（変更は削除、追加を含む。）する操作と、

（ｂ）前記（ａ）の当該単語の規定の変更の影響によって規定変更が必要となる可能性がある要件単語として、当該要件単語の変更前と変更後の生成第１リンク単語と被生成第１リンク単語とを摘出する操作と、

（ｃ）前記摘出された個々の単語について、その規定の変更が必要かどうかを検討する操作と、

（ｄ）変更の必要がある単語について前記（ａ）乃至（ｃ）の操作を繰り返す操作と

を備えることを特徴とする要件単語の変更方法。

【請求項２】ソフトウェアの要件として規定される要件単語（＝データ項目）群に係る、

該単語の名前、

該単語に対応する値を得る計算式（インプットによって値が得られることを含む。）、

該単語に対応する値が成立するための条件、
該単語がインプットかアウトプットかの別、
該単語の存在する記録媒体
で規定される情報を、規定された要件単語の配列の順序を問わないで自動プログラミングができる手法に適用して自動的に作成されるプログラムの新規開発においては、新規要件規定作業とその修正作業との両方が必要であるので、プログラムの新規開発にあたり、新規の要件規定は何もない状態からの変更と考えて、

（ａ）変更すべき要件単語自身の規定を変更（変更は削除、追加を含む。）する操作と、

（ｂ）前記（ａ）の当該単語の規定の変更の影響によって規定変更が必要となる可能性がある要件単語として、当該要件単語の変更前と変更後の生成第１リンク単語と被生成第１リンク単語とを摘出する操作と、

（ｃ）前記摘出された個々の単語について、その規定の変更が必要かどうかを検討する操作と、

（ｄ）変更の必要がある単語について前記（ａ）乃至（ｃ）の操作を繰り返す操作と

を備えることを特徴とする要件単語の新規規定方法。

【発明の詳細な説明】

【発明の属する技術分野】

本発明は、ソフトウェア開発手法に係り、特に、連鎖式データ項目規定法による要件定義法についての要件単語の変更方法及び要件単語の新規規定方法に関する。

【従来の技術】

プログラムの変更は常に必要である。新規に開発するプログラムの場合、初めに業務システムやプログラムシステムが総て決まっていることはむしろ少ないので、プログラムはプログラミング過程で変更しながら作成して行かなければならないことは通常の現象である。また既に完成されたプログラムでも、業務環境や経営の方針の変化に応じて要件は常に変化しているので、それに応ずるプログラムの変更は常に必要である。

しかしながら従来、新規開発の途上にあるプログラムにおいても、或いは既に一度は完成された既存のプログラムにおいても、プログラムの変更は容易ではない。その根本原因は従来法では処理順序を人がプログラムによってコンピュータに指示しなければならないことである。その事情をもう少し詳しく説明する。

従来法のプログラムではその命令文の順序にしたがってデータ処理が実行される。したがって処理順序は人が命令文を書く順序によって或いは条件付分岐の行く先を書くことによって指示しているし指示しなければならない。このように処理順序が決められているので１つのプログラムの中の１つのデータ項目の値はそれが処理されるまでの前処理と当該の処理によって決まる。したがって１つのデータ項目の値は１つのプログラムの中の何処に書かれているかによって変わってくる。したがって前処理を理解してさえいれば１つのデータ項目は１つのプログラムの中で別の定義式を持ってもよいし別の値を持ってもよい。つまり「１つのデータ項目は１つのプログラムの中でどんな前処理をされたかに拘わらず同じ値を持つ」ということではないのである。結局、処理順序を人が決めてやらなければならない代わりにデータ項目の定義や値に自由度が出てきてしまうのである。だから１つのデータ項目について１つのプログラムの中で何処でどんな定義で使われているのかはプログラムを作った人以外にはわからない。作った人でも長くは覚えてはいない。

例えばプログラムの中の一部に

y=ax+b

y=y+1

y=y+1

y=y+1

という命令文があったとする。最後のyは最初のyに3を足したものとなる。即ちyは処理段階の何処にいるかによって異なる値を持ち得るのである。この従来法の性質はＣＯＢＯＬ，　C++，　Java（登録商標）等のプログラム言語の種類に拘わらず結局は同じである。

さて、この時y=ax+b をある段階からy=cz-dに変更したいとする。プログラムの中ではyはいろいろな処理方法を指示されていろいろのところに潜んでいる。

これらを正しく変更しなければならない。しかしながら困ることは、
プログラムは人が法則もなく勝手に作ることができるので、yがプログラム全体の何処にあるか分らない。
yの前処理を遡ってたどって理解しなければyの計算式をどう変更してよいか分らない。
yはax+bという形で表現されているかもしれないのでyだけを探しても十分とは言えない。
yを変更することによりその波及効果として他のプログラム（例えばxやzの計算式）も変更しなければならないかもしれない。
また条件を付加して選択分岐を行わせるという変更の場合、その変更に応じてプログラムの他の部分においても条件付選択分岐の形に変更しなければならない。
等であるがそれらの総てを見つけて正しく変更することは容易ではない。

結局従来法においてプログラムの変更作業が困難である理由は、１箇所の変更に応じて他に変更すべきものが膨大なプログラム命令文の何処に潜んでいるのか容易に分らないことである。その根本原因は，繰り返しになるが、
・要件をコンピュータが理解できるように「人」が個々の業務要素の処理方法(Users' logic) の処理順序(Control logic)をプログラムの形にして指示しなければならないこと、したがって
・プログラムは「人」が勝手に処理順序(Control logic)を考えて作ったものであるからプログラム全体の中に何が何処にどんな形で書かれているのか分らない。１箇所変えればそのことによるプログラム全体への影響も機械的には分らない、
ことである。

従来法のプログラムは変更について以上のような困難性を内在している。したがって、総てを正しく変更するためには、人がプログラムの中の総ての命令文を詳細にたどって変更すべきところを摘出して変更することしかないのである。勿論業務の性質からプログラムのこの部分には変更が及ばないはずであるとして詳細なレビューをしないこともできるが、理論的には必要な変更はプログラムのどこにでも及ぶということを認識しなければならない。

以上に述べたのは、要件の変更に応ずるプログラムの変更についての従来法の特徴であった。しかしそもそもプログラムを新規に作成するときの従来法の特徴についても述べなければならない。

従来法では、前述の通り、データの処理順序は人が逐一プログラムによってコンピュータに指示しなければならなかった。しかし、錯綜したシステムでは、総てのデータの処理順序を人がプログラムによってコンピュータに間違いなく指示することは容易なことではない。処理順序を正しく指示するためには当然全体システムの総てのデータとその流れを正しく理解して把握しなければならない。そのために、業務分析、業務デザイン、データフローダイアグラムの作成、プログラムデザイン、等々のプログラミングの前の準備が必要でありまたプログラミングの膨大な作業が必要となる。このためコストも時間も膨大に掛かるのである。

もし要件(Users' logic)即ち要件単語の定義から特定の基本法則にしたがって処理順序(Control logic)を人が指示しなくともそれが自動的に導き出されることにより自動的にプログラムを生成することができるなら、要件の変更によるプログラムの変更は、要件単語の変更が正しく行われる限り、自動的に行うことができる。そうすればプログラムの変更は極めて容易になる。この場合１つのプログラムの中の１つのデータ項目は１つの値しか持たないようにする。そうしなければそもそも自動プログラミングはできないであろう。連鎖式要件定義法と要件の変更方法は正にその特徴を持っている。

図３にその関係を図解する。

【特許文献１】
国際公開第９７／１６７８４号パンフレット
【特許文献２】
国際公開第９８／１９２３２号パンフレット
【特許文献３】
国際公開第９９／４９３８７号パンフレット
【特許文献４】
国際公開第００／７９３８５号パンフレット
【特許文献５】

国際公開第０２／４２９０４号パンフレット

【特許文献６】

特開２００２－２０２８８３号公報

【発明が解決しようとする課題】

従来のソフトウェア開発方法は、前述の通りコンピュータがデータの処理順序を自ら見出すことができないことが原因となり

・プログラミングの準備やプログラミングのための人による膨大な作業

・プログラムの1つの変更の影響がプログラム全体の何処に及ぶか不明なのでプログラムを広範囲に調べる膨大な作業

等の必要性が起き、その結果下記のような問題が起きている。

・ソフトウェア開発に膨大な手間、時間、コストが掛かる。

・エンドユーザが業務システムをデザインするためにはある程度のシステムエンジニアリングの知識が必要であることから、結局、システムエンジニアに過度に依存し勝ちになる。

・実務経験がなく業務の運営・成果責任はないシステムエンジニ　アがリーダーシップを取り勝ちになる。

・システムエンジニアが主導するので、エンドユーザが本当に望むソフトウェアができるまでにやり直しが多くなるし必要なタイミングを逸することも多い。

・要件の変更に応ずるプログラムの変更が非常に厄介であり、要　件の変更には膨大な人力、コスト、時間がかかる。

本発明は、前述のような従来の技術が有していた諸問題点を解決しようとするものであって、従来法のように先ず要件を総て定義し次にプログラミングを行うのではなく、要件の定義を初めから構築する作業それ自身を個々の要件単語の定義のみによって簡単にかつ効率的に行い、またその要件の変更に応ずる要件単語の規定の変更を簡単にかつ効率的に行うことが可能な、連鎖式データ項目規定法による要件定義法についての要件単語の変更方法及び要件単語の新規規定方法を提供することを目的とするものである。

また、新規に規定された要件単語群或いは変更された後の要件単語群を、単語の処理順序を問わないで自動的にプログラミングすることができる手法に適用す

れば、人は従来法のような前述の膨大な作業をする必要がない。要件単語をこの種の自動プログラミング手法に適用する発明については、平成１４年９月９日付特許出願番号特願２００２－２６２６７０にて出願済である。

【課題を解決するための手段】

上記課題を解決するために、請求項１に係る本発明は、ソフトウェアの要件として規定される要件単語（＝データ項目）群に係る、該単語の名前、該単語に対応する値を得る計算式（インプットによって値が得られることを含む。）、該単語に対応する値が成立するための条件、該単語がインプットかアウトプットかの別、該単語の存在する記録媒体で規定される情報を、規定された要件単語の配列の順序を問わないで自動プログラミングができる手法に適用して自動的に作成されるプログラムの要件の変更にあたり、（ａ）変更すべき要件単語自身の規定を変更（変更は削除、追加を含む。）する操作と、（ｂ）前記（ａ）の当該単語の規定の変更の影響によって規定変更が必要となる可能性がある要件単語として、当該要件単語の変更前と変更後の生成第１リンク単語と被生成第１リンク単語とを摘出する操作と、（ｃ）前記摘出された個々の単語について、その規定の変更が必要かどうかを検討する操作と、（ｄ）変更の必要がある単語について前記（ａ）乃至（ｃ）の操作を繰り返す操作とを備えることを特徴とする。

ここで、或る単語の「生成第１リンク単語」とは、その単語の規定に含まれる他の総ての単語、即ち、その単語を生成するために必要な他の総ての単語のことをいう。

また、或る単語の「被生成第１リンク単語」とは、その単語が規定に含まれる他の総ての単語、即ち、その単語によって生成させられる他の総ての単語のことをいう。

さらに請求項２に係る本発明は、ソフトウェアの要件として規定される要件単語（＝データ項目）群に係る、該単語の名前、該単語に対応する値を得る計算式（インプットによって値が得られることを含む。）、該単語に対応する値が成立するための条件、該単語がインプットかアウトプットかの別、該単語の存在する記録媒体で規定される情報を、規定された要件単語の配列の順序を問わないで自動プログラミングができる手法に適用して自動的に作成されるプログラムの新規

開発においては、新規要件規定作業とその修正作業との両方が必要であるので、プログラムの新規開発にあたり、新規の要件規定は何もない状態からの変更と考えて、（ａ）変更すべき要件単語自身の規定を変更（変更は削除、追加を含む。）する操作と、（ｂ）前記（ａ）の当該単語の規定の変更の影響によって規定変更が必要となる可能性がある要件単語として、当該要件単語の変更前と変更後の生成第１リンク単語と被生成第１リンク単語とを摘出する操作と、（ｃ）前記摘出された個々の単語について、その規定の変更が必要かどうかを検討する操作と、（ｄ）変更の必要がある単語について前記（ａ）乃至（ｃ）の操作を繰り返す操作とを備えることを特徴とする。

上記で規定される本願に係る発明では、要件変更に応じてプログラムを変更（ここでは新規に開発するときを含めて考える）するにあたり、データ項目の定義や値についての自由度が出てくる余地を持つ従来法ではプログラムの中の総ての命令文を詳細にたどって変更すべきところを、機械的に摘出して変更するので、結局は総てを正しく変更することが可能となる。

またさらに、本願は、その本質上、上記で規定される要件単語に係る諸情報のみを、規定された単語（データ項目）の配列の順序を問わないで自動プログラミングができる手法に適用することで、プログラムを自動的に作成するプログラムの開発方法、或いは、プログラムの改変方法、プログラムの保守方法として捉えることも可能である。

【発明の概要】

既に特許出願している連鎖式要件定義法（特願２００２－２６２６７０）は新規に要件単語を規定する方法である。今回それをベースにして要件単語の変更のための効率的な方法を発明した。

連鎖式要件定義法は、ソフトウェアの要件を要件単語（＝データ項目）群の規定のみで定義する方法である。夫々の要件単語は、

単語の名前、

単語の値を得る計算式（インプットによって値が得
られることを含む。）、

単語の値が成立するための条件、

単語がインプットかアウトプットかの別、
単語の存在する記録媒体
だけを規定すればよい。

連鎖式要件定義は、意図を表すデータ項目から出発してまずその意図を導き出すデータ項目を規定し新たに現れたデータ項目を更に規定する。これを繰り返して次々に必要なデータ項目を引きずり出し、引きずり出されたデータ項目が総てインプットデータになるまで規定を続ける。この方法によって総てのデータ項目を引きずり出して規定する。それを要件定義とする。

従来法では、上記「従来の技術」の項で述べたように、データの処理順序は人がプログラムによって逐一コンピュータに指示しなければならなかった。そのため全体システムの総てのデータとその流れを正しく理解して把握しなければならない。そのためにプログラミングの前に、業務分析、業務デザイン、データフローダイアグラムの作成、プログラムデザイン、等々の準備が必要でありまたプログラミングの膨大な作業が必要となる。このためコストも時間も膨大に掛かるのである。

これに対して連鎖式要件定義法では、データの処理順序とは関係なくデータ項目を逐次的に規定して要件定義とするので、従来法において全部のデータ項目を初めに総て把握しそれらの処理順序を決めるための上記の厄介な準備作業が必要なくなり、非常に高い効率で要件定義ができるのである。

連鎖式要件定義法においてこのようにして規定された個々のデータ項目については、1つのプログラムの中で一義的に規定されているので１つのデータ項目は１つのプログラムの中の各所で使われても１つの値しか持たないことが保証されている。連鎖式要件定義法のこの特徴は従来法のプログラミングと根本的に異なる点である。上記「従来の技術」の項の例では連鎖式要件定義法では

$y=ax+b$

$y_1=y+1$

$y_2=y_1+1$

$y_3=y_2+1$

として別々のデータ項目として規定するのである。

このようにして引きずり出され規定されたデータ項目の順番は、データの処理順序とは全く異なるので、処理順序(Control logic)を人が指示しなければならない従来のプログラム開発手法では、このような要件定義は直接的には有効でない。しかし、処理順序不問の手法（例えばＬｙｅｅ（登録商標）。特許文献１乃至６参照）を用いればこの要件定義でも直接自動プログラミングができるので連鎖式要件定義法は完全に有効になり、要件定義法として連鎖式の高効率が生きることになる。

この方法において要件の変更とは上述の要件単語の幾つかについてその規定を変更することであるがそれはどのようにすればよいのであろうか。

一つ一つの要件単語を正しく変更するためには、個々の要件単語（例えば要件単語ａ）の規定の変更をするのみならず、その変更が他の要件単語の規定に与える影響についても検討しながら必要な変更をしなければならない。従来法ではそれが最も難しいのである。連鎖式要件定義法においてその変更方法を例を用いながら説明する。

変更すべき要件単語を単語aとする。変更前の単語aの定義としては、

a=b+c

成立条件：d<100

a=e-f

成立条件：d>=100

であるとする。

単語ａの定義をまず変更し、次にその影響によって変更の可能性がある単語を摘出し、それらの個々の単語についてその定義を変更する必要があるかどうかを検討し、必要があれば変更する。その手順は次の通りである。図１（要件単語の規定変更の手順図）を参照されたい。

（ａ）先ず、変更する要件単語a自身の規定を変更する。（変更は削除・追加を含む。）

例えばa=g+hに変更したい場合或いは成立条件を変更したい場合、変更前の計算式や成立条件その他のこの要件変更に応じて必要な要件単語規定を変更する。

削除の場合にも変更前の要件単語の定義が変更（削除）され　ることである。

新たに追加される要件単語の場合には変更前の要件単語の変更はなく新たに規定するのみである。

（ｂ）次に、（ａ）の要件単語ａの規定の変更の影響によって規定　　変更が必要となる可能性がある要件単語として、要件単語ａ　　の変更前と変更後の生成第１リンク単語群（下記＊１参照）　　及び単語リストの中の被生成第１リンク単語群（下記＊２参　　照）を摘出する。

＊１　ある要件単語aの生成第１リンク単語とは、その要件単　語aの規定に含まれる他のすべての単語である。その単語ａを生成するために必要な他の総ての単語である。例えば要件単語aの生成第１リンク単語とは上記のb, c, d, e, fの単語である。

＊２　ある要件単語ａの被生成第１リンク単語とは、その要件単語ａが規定に含まれる他のすべての単語である。その単語ａによって生成させられる他の総ての単語である。例えば要件単語ａの被生成第１リンク単語とは

ｐ＝ａ＊ｄ　とか

ｑ＝ｇ＋ａ

或いは

ｒ＝ｆ／ｈ　成立条件：ａ＜ｃ

等の場合におけるａを含むｐ，ｑ，ｒ単語である。

（ｃ）上記（ｂ）で摘出された個々の単語について、その規定の変更が必要かどうかを検討する。

（ｄ）上記（ｃ）の結果、変更の必要がある単語について上記の（ａ）、（ｂ）、（ｃ）と全くの同様の操作（Ｙオペレーション…図１参照）を繰り返す。

（ｅ）上記（ｄ）の結果、ある要件単語aから派生する総ての単語の規定に変更の必要がなくなればある要件単語aの変更操作は完了する。

ある要件単語ａに変更がなければ、その単語の生成第１リンク単語群にも被生成第１リンク単語群にも変更はなく、要件単語ａが原因となる波及的変更をする

必要がない。そこで変更の影響は遮断される。

何故変更の影響がその単語で遮断されるかというと、元々連鎖式要件単語の定義は１つの業務要素の規定を基にして次の業務要素を引きずり出す方法であるから、引きずり出す要素業務が変更前と同じであり変更がなければ、それ以降にその変更がなかった業務要素によって引きずり出される要素業務は変更前と同じになるのは自明だからである。もし引きずり出された業務要素に変更が必要ならそれは別件の新たな変更であり波及効果ではない。一方またある単語ａが他の単語の定義の中に使われていた場合、その単語ａに変更がないときには、他の単語も単語ａの波及効果としての変更はないことも自明である。したがって１つの要件単語の変更の影響は、変更がなくなった要件単語で遮断される。

以上の通り、本発明の連鎖式要件定義法における要件の変更方法は、１つの変更の影響が及ぶ範囲を限定でき、また変更の可能性のあるのはどんな要件単語であるかを提示する。これは総ての要件単語が連鎖式に生成されたことによって、互いに整合性を持った単語のネットワークを形成しており、したがって１つの単語は１つの値しか持たないことが保証されているからである。これは従来法のプログラムの変更とは根本的に違う点であり、遥かに優れているポイントである。この方法と処理の順序性を問わない自動プログラミング方法とがあいまって、ソフトウェアの開発・保守方法に劇的変化をもたらすものである。要件変更とプログラム変更が厄介な従来法の問題を根本的に解決するものである。従来法は前述の通り、１つのプログラムの変更に応じて他の部分で変更すべきプログラムがどこにあるのか、また、どの様に変更すべきかは人が考えなければならない。人が勝手に法則もなくプログラミングしたのでそれを見出すことは大変厄介なことである。理屈の上では変更の可能性はプログラム全体の何処にでもあるのでプログラム全体をしらみつぶしに探して考えなければならないのである。

上記の「発明の概要」の項で述べたことは、要件の変更方法であるが、要件を新規に定義するとき、変更すべき要件単語はない。しかし新規に要件単語を規定することは、何もない状態からの追加としての変更であるから、上記「発明の概要」の項で述べたことは新規の要件定義の方法でもある。したがって、要件単語の新規規定の場合にも、上記の（ａ）、（ｂ）、（ｃ）、（ｄ）、（ｅ）を行え

ばよいのである。

【発明の実施形態】

本願の発明の実施の具体的形態について、以下まず（１）で連鎖式要件定義法を簡単な例を用いて説明する。次いで（２）で図を用いて一般的な原理を説明する。更に（３）でこれらを実行するためのツールであるコンピュータソフトウェアについて説明する。

（１）例えば販売金額を求めるソフトウェアを開発しようとするとき、先ず求めるアウトプットである販売金額という単語（データ項目）を下記の式のように規定する。次にその計算式に現れた新しい単語（販売単価と販売数量）を夫々に規定する。同様のことを各単語がインプットデータと規定されるまで続け総てそうなれば完了する。

① 販売金額＝販売単価×販売数量、

② 販売単価＝仕入れ単価×（１＋マージン）、

③ 販売数量＝インプットデータ、

④ 仕入れ単価＝インプットデータ、

⑤ マージン＝インプットデータ、

因みにこれらの正しい処理順序は、個々の単語の値が得られるためにはその前にそれに必要な単語の値が得られていなければならないから、正しい処理順序について人は③、④、⑤、（この３つについては総てインプットデータであるからそれらの間の処理順序は問わない）、②、①の順であることを人は探り当てることができる。単語の数が多く、且つ、条件分岐も多い複雑なシステムでも、時間がどれほど掛かってもよければ、人は正しい順序を探り当てることは可能である。しかし残念ながら現在のソフトウェア開発手法では、コンピュータは処理順序を探り当てる機能を持っていない。「従来の技術」の項に述べたように、現在のソフトウェア開発の諸問題は、コンピュータのこの処理順序探索不能性から来ているのである。

（２）本願発明の原理を一般的に述べれば次の様になる。

ソフトウェアを開発したいときは必ず、ある求めたい単語（データ項目）が存在する。本願発明に係る連鎖式要件定義法は、一般的に、まずこの目的の単語を

決め、次にその単語の規定をし、次いでその規定式や成立条件の中に新たに出てきた新しい単語を更に規定する。これを連鎖的に次々に行う。最終的に新たな単語の規定がインプットデータとなれば完了する。夫々の単語の規定は、計算式、成立条件、インプット／アウトプットの別、その単語の存在する記録媒体の夫々の規定によって行う。

図２（連鎖式要件定義法）に単語間の関係の例を示す。

即ち、図２は、単語間の関係を模式的に示すと共に、連鎖式要件定義法（Data Chain Requirement Definition（DCRD)法と呼ぶ）を概念的に説明するための概念図である。なお、この図は単語（データ項目）間の関係を示すものであって、データ処理の順序を示すデータフローチャートとは全く異なることに注意されたい。

同図に示す例にあっては、目的データ項目Ａ（図中の１）は、中間データ項目Ｂ（図中の１１）、中間データ項目Ｃ（図中の１２）及び中間データ項目Ｄ（図中の１３）により規定されることを示している。さらにその中間データ項目Ｂ（図中の１１）は、Ｅファイル１１１中に格納されるインプットデータ項目Ｅ及びＦファイル１１２中に格納されるインプットデータ項目Ｆにより規定されることが示される。同様に、中間データ項目Ｃ（図中の１２）はＧ画面１２１上に存在するインプットデータ項目Ｇ及びＨ画面１２２上に存在するインプットデータ項目Ｈにより規定され、一方中間データ項目Ｄ（図中の１３）はＨ画面１２２上に存在するインプットデータ項目Ｈ及びＩファイル１３１中に格納されるインプットデータ項目Ｉにより規定されることが、それぞれ示されている。このようなデータ項目間の関係は、通常、種々異なる被開発ソフトウェアの種類、機能、コンピュータ構成等に応じて変化する。Ｅファイル１１１、Ｆファイル１１２、Ｉファイル１３１及びＧ画面１２１並びにＨ画面１２２は事実上オプションであり、また、ここには図示されないこれら以外の媒体上にインプットデータ項目が登載されていてもよい。

上記のような図２の状況下において、更に単語（データ項目）間の関係の中で次のような条件があるとする。即ち、

もし　Ｆ＞Ｇなら、　　Ａ＝Ｂ×Ｃ／Ｄ

そうでなければ、　　Ａ＝Ｃ＋Ｇ＋Ｈ

因みに上記を従来法の一般的プログラム言語で書けば、

if　Ｆ＞Ｇ　then　Ａ＝Ｂ×Ｃ／Ｄ

else　　Ａ＝Ｃ＋Ｇ＋Ｈ

図２のような単語（データ項目）間の関係と上記の条件設定がある場合、連鎖式要件定義法による要件定義は次のようになる。

Ａ　＝　Ｂ×Ｃ／Ｄ　（定義式の規定）

Ｆ＞Ｇ　（成立条件の規定）

アウトプットデータ。（インプット／アウトプットの別の規定）

Ａ画面にある。（存在する記録媒体の規定）

（ここでの新しいデータ項目はＢ、Ｃ、Ｄ、Ｆ、Ｇだから夫々下記に規定する。以下新しいデータ項目が出てくる毎に規定を行う。）

Ａ＝Ｃ＋Ｇ＋Ｈ

Ｆ＜＝Ｇ

アウトプットデータ。

Ａ画面にある。

Ｂ　＝　Ｅ＋Ｆ

Ｃ　＝　Ｇ／Ｈ

Ｄ　＝　Ｈ－Ｉ

Ｅ　＝　インプットデータ。Ｅファイルにある。

Ｆ　＝　インプットデータ。Ｆファイルにある。

Ｇ　＝　インプットデータ。Ｇ画面にある。

Ｈ　＝　インプットデータ。Ｈ画面にある。

Ｉ　＝　インプットデータ。Ｉファイルにある。

実際の作業においては、例えば、下記の表１に示すような要件定義のための単語規定作成表を用いて、新たに現れる各単語に対して規定を記述するようにすれ

ば、漏れなく所望の効果を達成することができる。

【表１】

連鎖式要件定義(Data Chain Requirement Definition)のための単語規定作成表

データ項目番号	データ項目呼称	データ項目ID	計算式	成立条件	I	O	記録媒体呼称	記録媒体ID	備考

（３）連鎖式要件定義法によって実際に要件単語を新規に規定する場合或いは要件の変更に応じて要件単語を変更する場合、それを実行するためのツールであるコンピュータソフトウェアを使うとよい。

ツールソフトの基本原理は極めて簡単である。ここでもう一度確認するが、要件の変更とは「要件単語の定義即ち個々の要件単語の規定の変更」でありその変更・削除・追加である。ツールソフトの基本原理は本発明の基本原理と当然同じであり下記の通りである。

（ａ）　変更する要件単語の変更前の定義の画面表示（形式は表１）とそれに対する変更操作

（ｂ）　変更する単語の変更前と変更後の生成第１リンク単語群
と被生成第１リンク単語群の定義の画面表示
（この（ａ）と（ｂ）が前記のＹオペレーションである。）

（ｃ）　上記（ｂ）の個々の単語について、規定変更が必要かどうかの人による検討

（ｄ）　上記（ｃ）の結果、変更の必要がある単語について上記の（ａ）、（ｂ）、（ｃ）と同様の操作

（ｅ）　上記（ｄ）の結果ある或要件単語から派生する総ての単語の規定に変更の必要がなくなれば当該要件単語の変更操作は完了する。

変更が要件単語の新規追加の場合は、新規にその定義をすることは規定なしの状態から規定ありの状態への変更であるから、新規定義操作として（ａ）の変更操作を行えばよく、次にそれに応じて（ｂ）、（ｃ）、（ｄ）、（ｅ）を行えばよい。

全くの新規に要件を定義するために要件単語の定義をしていく場合は変更すべき要件単語がいまだ存在しない。その定義をするためには新規定義操作として何もないものからの上記（ａ）の変更操作を行えばよく、次にそれに応じた生成第１リンク単語群の定義を（ｂ）、（ｃ）、（ｄ）、（ｅ）で新規定義操作を行えばよい。

以上から、ツールソフトとしては変更のためのものも新規開発のためのものも全く同じものである。元々新規開発の場合も、初めから要件定義が全部確定している訳ではなく個々の要件定義を試行錯誤的に変更を繰り返しながら行っていく間に全体の要件定義が構築されるのが現実である。したがってその途中の変更は本来避けられない。その観点から新規開発向けも変更向けも同じツールであるべきである。そしてその頻繁な変更が容易に行えるならばソフトウェア開発としては大変望ましい。その能力を持つ連鎖式要件定義法のツールソフトはソフトウェア開発方法としてその生産効率を飛躍的に高める。

本発明の多くの効果および利点は明細書の詳細な説明から明白である。これまでに説明してきた効果及び利点を要約すると次のようになる。

上記「従来の技術」の項で述べたように、プログラムの変更は常に必要である。既に完成されたプログラムの場合でも、業務環境や経営の方針の変化に応じて要件は常に変化しているので、それに応ずるプログラムの変更は常に必要である。

また新規に開発するプログラムの場合でも、初めに業務システムやプログラムシステムが総て決まっていることはむしろ少ないので、プログラムはプログラミング過程で変更しながら作成して行かなければならない。

一方、上記「従来の技術」の項で述べたように、従来法のプログラムではその命令文の順序にしたがって計算が実行される。したがって処理順序は人が命令文を書く順序によって或いは条件付分岐の行く先を書くことによって指示している

し指示しなければならない。そうすると処理順序が決められているので１つのプログラムの中の１つのデータ項目の値はそれが処理されるまでの前処理と当該の処理によって決まる。したがって１つのデータ項目の値は１つのプログラムの中の何処に書かれているかによる。したがって前処理を理解してさえいれば１つのデータ項目は１つのプログラムの中で別の計算式を持ってもよいし別の値を持ってもよい。つまり１つのデータ項目が１つのプログラムの中でどんな前処理をされたかに拘わらず同じ値を持つということは保証されていないのである。結局、処理順序を人が決めてやらなければならない代わりにデータ項目の定義や値に自由度が出てきてしまうのである。だから１つのデータ項目は１つのプログラムの中で何処でどんな定義で使われているのかは作った人以外にはわからない。作った人でも長くは覚えてはいない。

このような状況において、要件変更に応じてプログラムを変更するとき、1つのプログラムの変更がプログラムの他の部分に与える影響を考え必要ならばそれらを正しく変更しなければならない。そのために従来法では上記のようにデータ項目の定義や値に自由度が出てくるためにプログラムは人が法則もなく勝手に作ることができるから、必要な変更がプログラム全体の何処にあるか分らなくなってしまうのである。必要な変更の総てを見つけて正しく変更することは容易ではない。１箇所変えればそのことによるプログラム全体への影響は機械的には分らないからである。総てを正しく変更するためには、人がプログラムの中の総ての命令文を詳細にたどって変更すべきところを摘出して変更することしかないのである。

本発明の連鎖式要件定義方法は、要件の定義においても要件の変更においても従来法の上記の問題を総て解消してしまうのである。

さらにこの連鎖式要件定義法によって定義された要件単語群を、処理順序を問わない自動プログラミング法（例えばＬｙｅｅ（登録商標）。特許文献１乃至６参照）に適用すれば、ソフトウェア開発が新規開発においても要件の変更においても、要件の定義過程の端緒からプログラミングに至るまで人の考えなければならないことを極小化して自動化され、劇的な効果を得ることができる。

本発明は当該技術分野における通常の知識を有する者にとって修正および改変

が容易に数多くなし得るので、図示および記述されたものと寸分違わぬ構成および動作に本発明を限定することは望ましくないことであり、従って、あらゆる適切な改変体および等価体は本発明の範囲に含まれるものと見なされうる。前述の本発明に係る実施の具体的形態の説明および例示によって詳細に記述されたが、本願の特許請求の範囲のみならず本発明に係る開示事項全体に定義された本発明の範囲から逸脱することなしに、修正、置換、および、変更が数多く可能である。

また、本願に係る発明は、その適用において、上記の記述において説明されるか、或いは、図面に示された要素の詳細な解釈及び組み合わせに限定されるものではない。本発明は、他の実施形態が可能であり、種々の方法で実用および実施可能である。また、ここで用いられた語法および用語は記述を目的とするものであり、限定的に働くものとみなされてはならない。

従って、当該技術分野における通常の知識を有する者は、本開示の基調となる概念は、本発明の幾つかの目的を実施するための他の構造、方法、及び、システムを設計するための基礎として容易に利用され得ることを理解するはずである。従って、本発明の趣旨および範囲から逸脱しない限り、本願の特許請求の範囲にはそのような等価な解釈が含まれるものと見なされるものである。

また、本発明に係る技術思想は、例えばコンピュータソフトウェアの自動開発装置、自動開発プログラム、自動開発プログラムを記録した記録媒体、伝送媒体、紙媒体としても、また、自動開発プログラムを登載したコンピュータ・装置、自動開発プログラムを実行するクライアント・サーバ形式等といったカテゴリーにおいても実現、利用可能であることはいうまでもない。

さらに本発明は、単一プロセッサ、単一ハードディスクドライブ、及び、単一ローカルメモリを備えたコンピュータシステムに限らず、当該システムのオプションとして、任意の複数または組み合わせプロセッサ又は記憶デバイスを装備するにも適している。コンピュータシステムは、精巧な計算器、掌タイプコンピュータ、ラップトップ／ノートブックコンピュータ、ミニコンピュータ、メインフレームコンピュータ、及び、スーパーコンピュータ、ならびに、これらの処理システムネットワーク組合わせを含む。本発明の原理に従って作動する任意の適切

な処理システムによって代替されうるし、また、これらと組合せて用いることも可能である。

また、本発明に係る技術思想は、もとよりあらゆる種類のプログラミング言語に対応可能である。さらに本発明に係る技術思想は、あらゆる種類・機能のアプリケーションソフトウェアに対しても適用可能である。

またさらに本願発明は、その技術思想の同一及び等価に及ぶ範囲において様々な変形、追加、置換、拡大、縮小等を許容するものである。また、本願発明を用いて生産されるソフトウェアが、その２次的生産品に登載されて商品化された場合であっても、本願発明の価値は何ら減ずるものではない。

【発明の効果】

以上、詳細に説明したように、本発明の連鎖式要件定義方法によれば、要件の新規定義においても要件の変更においても従来法の「1つのプログラムの変更の影響によって必要となる要変更箇所がプログラム全体の何処にあるか分らなくなってしまう」という問題を総て解消してしまうのである。

さらにこの連鎖式要件定義法によって定義された要件単語群を、処理順序を問わない自動プログラミング法（例えばＬｙｅｅ）に適用すれば、ソフトウェア開発が新規開発においても要件の変更においても、要件の定義過程の端緒からプログラミングに至るまで人の考えなければならないことを極小化して自動化され、劇的な効果を得ることができる。

【図面の簡単な説明】

【図１】

要件単語の規定を変更（追加を含む）するときに、1つの変更の影響によって派生する必要な変更も同時に行いながら変更して行く手順を示している概念図である。

【図２】

データ項目間の関係を模式的に示すと共に、連鎖式要件定義法（Data Chain Requirement Definition (DCRD)法）を概念的に説明するための概念図である。

【図３】

要件定義と目的プログラムとの関係について、従来技術と本願に係る発明とを

比較論的に説明するための概念図である。

【符号の説明】

１　　目的データ項目Ａ及び画面Ａ

１１　　中間データ項目Ｂ

１２　　中間データ項目Ｃ

１３　　中間データ項目Ｄ

１１１　　インプットデータ項目Ｅ及びＥファイル

１１２　　インプットデータ項目Ｆ及びＦファイル

１２１　　インプットデータ項目Ｇ及びＧ画面

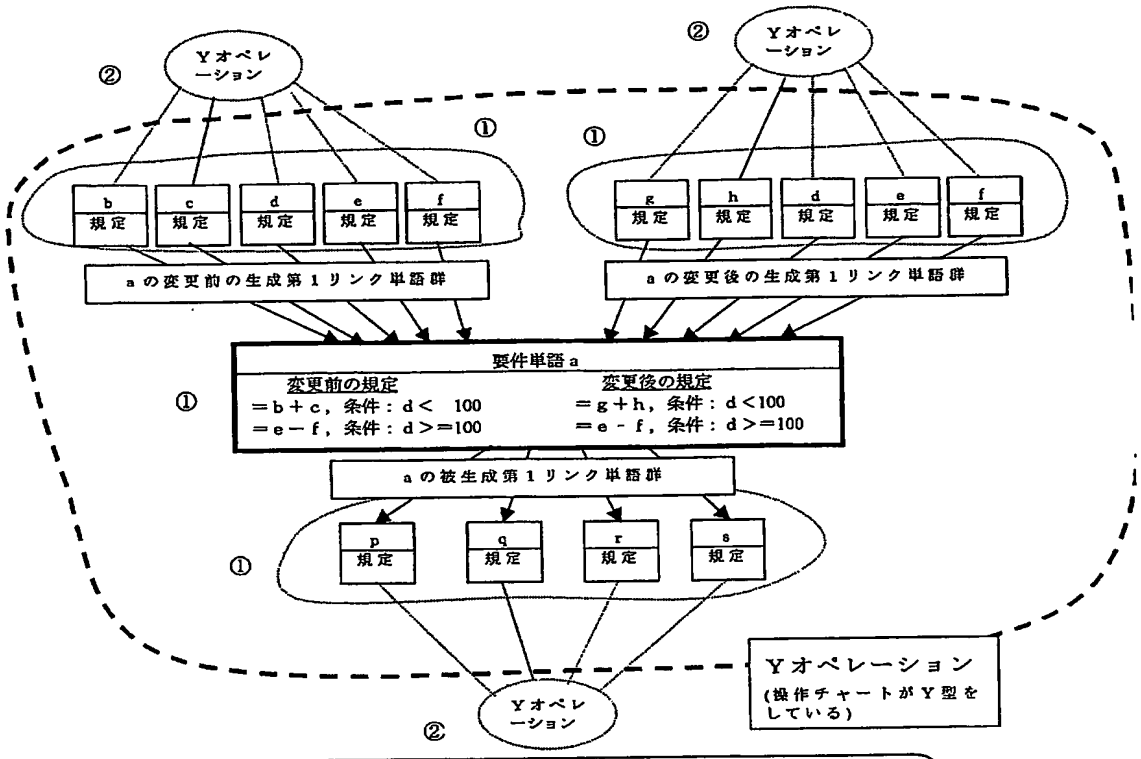
１２２　　インプットデータ項目Ｈ及びＨ画面

１３１　　インプットデータ項目Ｉ及びＩファイル

【書類名】　図面

【図１】

要件単語の規定変更の手順図（Ｙオペレーション）

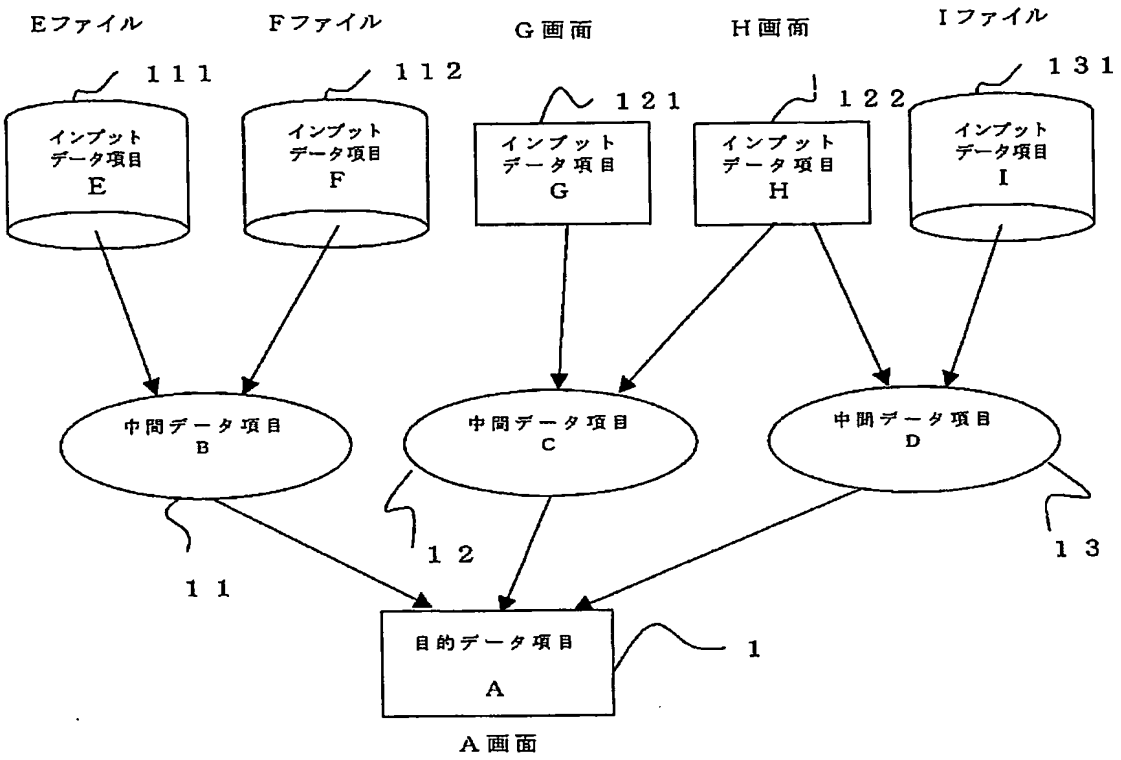


① この要件単語ａの規定を変更する。
変更前と変更後の生成第１リンク単語群を摘出する。
（ｂ，ｃ，ｄ，ｅ，ｆ，　と　ｇ，ｈ，ｄ，ｅ，ｆ）
この要件単語の被生成第１リンク単語群を摘出する。
（ｐ，ｑ，ｒ，ｓ）

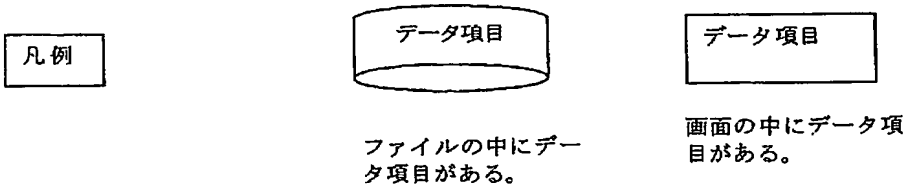
② 生成第１リンク単語群と被生成第１リンク単語群の個々の要件単語の規定に変更が必要かどうか検討する。
変更が１つでもあれば、その当該個別単語についてＹオペレーションを行う。
総てのリンク単語に変更がなくなるまで続ける。一つもなければそれ以降のリンク単語に変更は波及しないので終了する。

【図２】

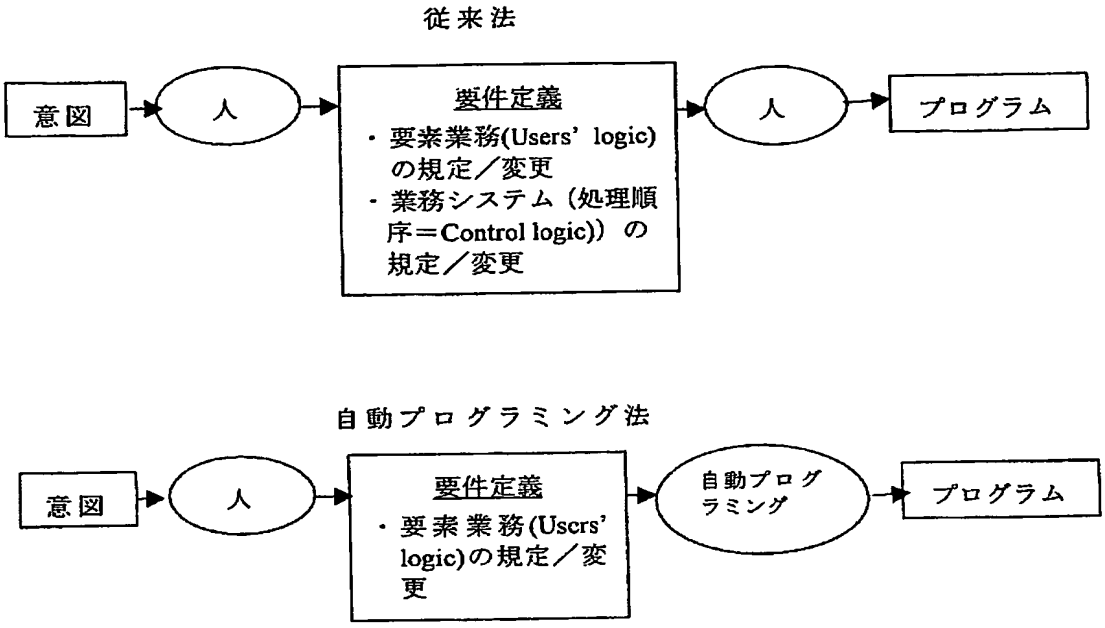
連鎖式要件定義法
(Data Chain Requirement Definition (DCRD)法)
データ項目間の関係図



上図のデータ項目Ａ，Ｂ，Ｃ，Ｄの関係は、Ａというデータ項目はＢデータ項目，Ｃデータ項目，Ｄデータ項目から成ることを示す。



【図３】


従来法
意図
人
要件定義
・要素業務(Users' logic)の規定／変更
・業務システム（処理順序＝Control logic)）の規定／変更
人
プログラム
自動プログラミング法
意図
人
要件定義
・要素業務(Users' logic)の規定／変更
自動プログラミング
プログラム

【書類名】　要約書

【要約】

【課題】

個々の要件単語を規定することによって開発要件を定義する方法において、新規に要件を開発する場合も既存の要件を変更する場合も、要件定義の構築および変更の作業の効率を劇的に高めること。

【解決手段】

開発するソフトウェアが最終的に求めようとするアウトプット項目を取りだし、そのデータ項目の生成方法を定義し、その定義に新たに現れたデータ項目の生成方法を更に定義する。これを総ての新たなデータ項目の生成がインプットデータによることになるまで継続して完成する要件単語群を変更するにあたり、変更前と変更後の該単語の生成方法を定義する別の単語および他の単語のうち該単語によって生成方法が定義される単語とその定義を摘出して提示し、その変更の要否を検討し、「要」なら変更を継続し「否」なら影響はそこで遮断されるので該単語の変更は終了する（図１参照）。従来法において変更の影響の範囲限定不能問題が解決される。

【選択図】　図１

特願２００３－０１０４５４

出　願　人　履　歴　情　報

識別番号　　［５９９０８６２３８］

１．変更年月日　　１９９９年　６月２１日
［変更理由］　　新規登録
住　所　　東京都港区高輪３丁目１１番３号
氏　名　　ソフトウェア生産技術研究所株式会社

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☑ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.